Patent Application Cover Page

PROCESSING RULES FOR DIGITAL MESSAGES

Inventors:

W. Todd Daniell

Jeffrey R. Kuester
Sam S. Han
Thomas, Kayden, Horstemeyer & Risley LLP
100 Galleria Parkway
Suite 1750
Atlanta, GA 30339
Tel: 770.933.9500
Fax: 770.951.0933

Attorney Ref. No.: 190250-1580
BellSouth Ref. No.: BLS-030455

Customer No.: 38823

PROCESSING RULES FOR DIGITAL MESSAGES

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application incorporates by reference the following applications, as if they were set forth in their entireties: U.S. patent application having serial number 10/274,405, filed October 18, 2002; U.S. patent application having serial number 10/274,408, filed October 18, 2002; U.S. patent application having serial number 10/274,478, filed October 18, 2002; U.S. patent application having serial number 10/325,268, filed December 19, 2002; U.S. patent application having serial number 10/610,736, filed June 30, 2003; U.S. provisional patent application having serial number 60/411,336, filed September 17, 2002; U.S. provisional patent application having serial number 60/411,438, filed September 17, 2002; U.S. provisional patent application having serial number 60/416,916, filed October 8, 2002; U.S. provisional patent application having serial number 60/419,613, filed October 17, 2002; U.S. provisional patent application having serial number 60/426,145, filed November 14, 2002; U.S. provisional patent application having serial number 60/426,146, filed November 14, 2002; U.S. provisional patent application having serial number 60/426,422, filed November 14, 2002; U.S. provisional patent application having serial number 60/426,432, filed November 14, 2002; and U.S. provisional patent application having serial number 60/426,440, filed November 14, 2002.

[0002] Co-pending U.S. patent applications having serial number [attorney docket no. 190250-1300], titled "Identifying Undesired Email Messages Having Attachments," filed on October 14, 2003; [attorney docket no. 190250-1570], titled "Filtered Email Differentiation," filed on October 14, 2003; and [attorney docket no. 190250-1610], titled "Phonetic Filtering of Undesired Email Messages," filed on October 14, 2003, are also

incorporated herein by reference in their entireties.

FIELD OF THE DISCLOSURE

[0003] The present disclosure relates generally to electronic communications and, more particularly, to network communications.

BACKGROUND

[0004] Email clients have been used extensively as a digital communications medium between two parties. Email clients have incorporated rule-based processing in order to facilitate organization of incoming email messages. One such example of a rule-based processing system and method is provided in U.S. patent number 5,917,489 (hereinafter "the '489 patent"), by Thurlow *et al.*, which issued on June 29, 1999. In that system, a "rules wizard" is provided to an email user, thereby permitting the user to select various permutations of conditions, actions, and exceptions. Since the conditions, actions, and exceptions are described in detail in the '489 patent, further discussion of conditions, actions, and exceptions is omitted here.

[0005] While a "rules wizard" facilitates the organization of email messages, the functionality of the "rules wizard" is limited to the known subset of conditions, actions, exceptions, and various permutations thereof, which are defined for the particular email client. Additionally, the available set of rules is limited to processing email communications. Hence, those rules only provide organization mechanisms within the realm of email messages.

[0006] In view of the limitations of existing "rules wizards," a heretofore unaddressed need exists in the industry.

SUMMARY

[0007] The present disclosure provides for processing rules for digital messages.

[0008] Briefly described, some embodiments are directed to determining whether an email message meets a predefined condition, and executing an action in an instant messaging (IM) system in response to determining that the email message meets the predefined condition.

[0009] Other embodiments are directed to providing a programming interface, and storing inputs provided by a user at the programming interface. For those embodiments, the programming interface is adapted to receive user input in the form of a markup language. The inputs comprise a condition and an action.

[0010] Yet other embodiments are directed to determining whether a digital message meets a predefined condition, and executing a filtering algorithm on the digital message in response to determining that the digital message does not meet the predefined condition. The digital message may be, for example, an email message or an IM message

[0011] Other systems, devices, methods, features, and advantages will be or become apparent to one with skill in the art upon examination of the following drawings and detailed description. It is intended that all such additional systems, methods, features, and advantages be included within this description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Many aspects of the disclosure can be better understood with reference to the following drawings. The components in the drawings are not necessarily to scale, emphasis instead being placed upon clearly illustrating the principles of the present

disclosure. Moreover, in the drawings, like reference numerals designate corresponding parts throughout the several views.

[0013] FIG. 1 is a block diagram showing an embodiment of a system for processing rules.

[0014] FIG. 2 is a flowchart showing an embodiment of a method for processing rules.

[0015] FIG. 3 is a flowchart showing another embodiment of a method for processing rules.

[0016] FIG. 4 is a flowchart showing, in greater detail, the step of determining whether or not a digital message meets a predefined condition, which is shown in FIG. 3.

[0017] FIG. 5 is a flowchart showing yet another embodiment of a method for processing rules.

[0018] FIG. 6 is a flowchart showing a specific embodiment of another rules-processing method.

[0019] FIG. 7 is a diagram showing an example graphical user interface (GUI) for an embodiment of a system for processing rules.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0020] Reference is now made in detail to the description of the embodiments as illustrated in the drawings. While several embodiments are described in connection with these drawings, there is no intent to limit the disclosure to the embodiment or embodiments disclosed herein. On the contrary, the intent is to cover all alternatives, modifications, and equivalents.

[0021] In order to remedy some of the deficiencies of prior systems, various embodiments for processing rules are presented herein. In some embodiments,

integration of email and instant messaging (IM) are shown in the context of processing rules for digital messages. For example, in some embodiments, actions are performed in IM when a condition is met in email. In a specific example, when an email message is received from a particular sender, the system may determine whether that sender is present on the Internet, and automatically launch an IM chat session with the sender if that sender is present.

[0022] In other embodiments, a programming interface is provided so that a user may customize specific conditions and actions, rather than merely selecting various permutations of predefined conditions and actions. In this regard, greater flexibility is provided to the user. In a specific example, the programming interface may be amenable to user input in the form of a markup language, such as Hypertext Markup Language (HTML) or Extensible Markup Language (XML). Thus, if a user is sufficiently adept at programming in these languages, that user may vastly expand the content of the rules for processing digital messages.

[0023] In other embodiments, a filtering algorithm is integrated with the rule engine, thereby providing an additional layer of functionality. For example, an email application may be configured to perform a Bayesian filtering of all incoming email messages in the absence of an indication to the contrary. In other words, the email application combines both a user-definable rules-based approach and a standard-algorithm-based approach to filtering digital messages. In this regard, filtering power is improved by combining the two separate approaches. Greater details of such systems and methods are provided below.

[0024] FIG. 1 is a block diagram showing an embodiment of a system for processing rules. As shown in FIG. 1, some embodiments of email systems comprise workstations

172, 174, 176 that are coupled to a server 150 over a network such as the Internet 180. The server 150 is coupled to a database 162 that stores the email accounts (or mailboxes) of various users.

[0025] In the operating environment shown in FIG. 1, a sender of an email message generates the email message at a sender workstation 172 and sends the email message through a network 180 (which may include a server 150 and a database 162) to a recipient at a recipient workstation 176. As shown in FIG. 1, the recipient workstation 176 includes a processor 182, a network interface 190, a memory 184, a local storage device 188, and a bus 186 that permits communication between the various components.

[0026] While not explicitly shown, it should be appreciated that the other workstations 172, 174 may also include similar components that facilitate computation or execution of applications on the workstations 172, 174. In some embodiments, the local storage device 188 may be a hard drive configured to electronically store data. The local storage device 188 may also store computer programs that execute on the recipient workstation 176. In this sense, the processor 182 is configured to access any program that is stored on the local storage device 188, and execute the program with the assistance of the memory 184.

[0027] In the embodiment of FIG. 1, an email application 181 is shown as being loaded into memory 184 for launching at the workstation 176, thereby permitting the workstation 176 to send and receive email messages through the network 180. Additionally, the memory 184 is shown as having an instant messaging (IM) application 183, which permits users at the workstation 176 to send and receive IM messages over the network 180. Moreover, a programming interface 185 and command execution logic 187 are shown as being loaded into memory 184. As described in greater detail below, the

programming interface 185 and the command execution logic 187 are configured to provide the relevant functionality for extending conventional rule engines. Since the several embodiments below are described in conjunction with email and IM, it should be appreciated that both the programming interface 185 and the command execution logic 187 may be coupled to the email application 181 and the IM application 183. In this regard, both the email application 181 and the IM application may separately access the programming interface 185 and the command execution logic 187 in order to establish processing rules for incoming and/or outgoing digital messages.

[0028] Since the functioning of computing devices is well known in the art, further discussion of the processor 182, the memory 184, and the local storage device 188 are omitted here. However, it should be appreciated that the memory 184 may be either volatile or non-volatile memory.

[0029] The network interface 190 is configured to provide an interface between the recipient workstation 176 and the network. Thus, the network interface 190 provides the interface for the workstation 176 to receive any data that may be entering from the network and, also, to transmit any data from the workstation 176 to the network. Specifically, in some embodiments, the network interface 190 is configured to permit communication between each of the workstations 172, 174, 176 and the server 150 and, additionally, to permit communication among the workstations 172, 174, 176 themselves. In this regard, the network interface 190 may be a modem, a network card, or any other interface that interfaces each of the workstations 172, 174, 176 to the network. Since various network interfaces are known in the art, further discussion of these components is omitted here. It should be understood that various aspects of the email application 181 may be conventional or may be custom tailored to specific needs.

[0030] Similar to the workstation 176, the server 150 may also include a processor 152, a memory 154, a network interface 160, and a local hard drive 158, which are in communication with each other over a local bus 156. Since the components 152, 154, 156, 158, 160 at the server 150 perform largely similar functions as the components 182, 184, 186, 188, 190 at the workstation 176, further discussion of the server-side components is omitted here.

[0031] An example of a conventional rule engine is described in U.S. patent number 6,057,841 (hereinafter "the '841 patent"), issued to Thurlow *et al.* and assigned to Microsoft® Corporation. The '841 patent is incorporated herein by reference as if set forth in its entirety. Unlike the '841 patent, the embodiments below provide integration between email and IM. Since systems and methods for integrating email and IM are described in greater detail in U.S. patent application serial number 10/325,268 and U.S. patent application serial number 10/274,408, only a truncated discussion of the integration of IM and email is provided here. By integrating IM and email as taught in U.S. patent application serial number 10/325,268 and U.S. patent application serial number 10/274,408, the universe of rules in the '841 patent may be extended from the closed set of rules, which only relate to email, to a vaster set of rules, which encompasses both email and IM. Example embodiments of rules that encompass both email and IM are shown with reference to FIGS. 5 and 6.

[0032] Another distinction is that, unlike the '841 patent, various embodiments of the present disclosure integrate a filtering algorithm in conjunction with a rules-based approach. Thus, while the '841 patent operates in a closed set of predefined rules, some embodiments of the present disclosure supplement the set of rules with additional filtering processes, such as, for example, Bayesian filters. In this regard, a more powerful filtering

engine is provided to the email user. Since additional filtering algorithms, such as Bayesian filters, are described in greater detail in 10/610,736, filed on June 30, 2003, [attorney docket no. 190250-1300], titled " Identifying Undesired Email Messages Having Attachments," and [attorney docket no. 190250-1610], titled "Phonetic Filtering of Undesired Email Messages," further discussion of additional filtering algorithms is omitted here. Example embodiments having combined rules and filtering algorithms are provided with reference to FIGS. 3 and 4.

[0033] Yet another distinction between the '841 patent and the various embodiments described herein is that, unlike the '841 patent, the embodiments of the inventive email and IM applications provide a programming interface 185 that permits expansion of the set of rules. In other words, the '841 patent only provides a limited set of conditions, actions, and exceptions from which the user may select various permutations. To the contrary, the programming interface 185, described in greater detail below, provides a user interface in which conditions, actions, and exceptions may be customized or programmed directly by the user. In this regard, the user may exponentially extend the set of rules to accommodate almost every need. Example embodiments that provide programming interfaces are shown with reference to FIG. 2.

[0034] Also, unlike the '841 patent, which stores all of the condition, actions, and exceptions in a proprietary language and links these with the mail application programming interface (MAPI) and operating system, various embodiments of this disclosure store the conditions, actions, and exceptions using a markup language, such as, for example, Hypertext Markup Language (HTML) or Extensible Markup Language (XML). In this regard, rules in some of the embodiments of this disclosure are portable to other operating systems and environments. An example XML-based rule engine may be

configured to perform one or more actions when an email message is received, and the email message matches one or more conditions defined by the rule. In some embodiments, the rule engine may be developed using Microsoft Visual C++ 7.0 and the Active Template Library (ATL) version 7.0, in accordance with known methods. Since one example of an acceptable mechanism for discerning whether an email message matches a condition is described in great detail in the '841 patent, further discussion of that mechanism is omitted here.

[0035] In some embodiments, the data required to define a rule may include a rule identifier (ID), a rule type, a condition (also referred to herein as a "rule criterion" or, simply, "criterion"), and an action (also referred to herein as "rule action").

[0036] The rule ID uniquely identifies each rule. In this regard, a new rule ID is assigned to each newly-created rule. Preferably, the rule ID is assigned by the system and, upon assignment, maintained and tracked by the system using, for example, a database or a lookup table. In a preferred embodiment, the rule ID is a text representation of a 6-digit number used to identify a rule.

[0037] The rule type identifies the origin of the rule, and is designed to determine the source and/or purpose of the rule. In some embodiments, the rule type may include system rules, personal rules, SPAM rules, and parental control rules (also referred to as "child" rules).

[0038] The system rules are preferably rules that may be defined by the vendor of a particular email application or a particular IM application. In this regard, the system rules may be rules that are pre-packaged with the particular email or IM software.

[0039] The personal rules may be user-defined rules, which may be defined with the assistance of the particular email or IM application. In this regard, some personal rules

may be defined using a "rules wizard" somewhat similar to that described in the '841 patent. Other personal rules may be defined using the programming interface 185, which permits customized code writing by the user.

[0040] The SPAM rules relate to filtering algorithms that may be used in conjunction with system rules or personal rules. Thus, the SPAM rules may be invoked in response to a particular condition being met.

[0041] The child rules relate to parental control functionality. In this regard, the child rules may be accessible by users having predefined access levels. For example, if both a parent and a child share the same computer and email application, then the child rules may be invoked or disabled only by the parent. In this regard, the parent may prevent the child from disabling certain rules.

[0042] The rule criterion (or condition) is used to determine whether or not to apply a particular rule. In some embodiments, the rule criterion may include two parts: (1) rule criterion type; and (2) rule criterion data. In other words, if the rule criterion is implemented in XML, then the rule criterion may have an XML tag as the criterion type and an argument associated with the XML tag as the rule criterion data. The following CHART 1 provides, among others, example rule criterion types, their corresponding rule criterion data, and the description of the criterion data. The rule criterion types are identified by their corresponding XML tags.

Criterion Type	*TAG*	*Description*	*Criterion Data*
From Address	FROMADDR	Message is from a specific internet address.	Internet address
From Domain	FROMDOMAIN	Message is from a given internet domain.	Internet Domain Name
To Address	TOADDR	Message was sent to a specific internet address	Internet Address
Cc Address	CCADDR	Message was Carbon Copied to a specific internet address	Internet Address
Subject Keyword	SUBJECTKEY	Message Subject contains a keyword or keyword list	Keyword or keyword list
Body Keyword	BODYKEY	Message Body contains a keyword or keyword list	Keyword or keyword list
Body XML TAG	BODYTAG	Message Body contains an XML or HTML TAG	Tag Name
Empty Message Subject	NOSUBJECT	The Message Subject was empty	Nothing
Empty Message Body	NOBODY	The Message Body was empty	Nothing
Message Size greater than	MSGSIZE	The Message Body size was greater than a given size	Size in bytes
All Messages	ALL	All Messages	Nothing
If Sender is Presently online on BIM.	SENDERPRESENT	Is the sender currently logged into BIM and present?	Nothing
Source IP Address	SOURCEIP	The Message sent from a given source IP address	IP Address
Source IP Range	SOURCEIPRANGE	The Message was sent from a range of IP Addresses	IP Address , IP Address
Bayesian Filter Test	BAYESIAN	The message will be tested against the Bayesian Probability Engine to determine if this message is considered SPAM.	Nothing

CHART 1: Rule Criteria (Conditions)

[0043] The rule action is the action that will be performed if its corresponding condition is met. Similar to the rule criterion, the rule action may include two parts: (1) rule action type; and (2) rule action data. Thus, if the rule action is implemented in XML, then the rule action may have an XML tag as the action type and an argument associated with the XML tag as the rule action data. The following CHART 2 provides, among others, example rule action types, their corresponding rule action data, and the description of the action data. The rule action types are identified by their corresponding XML tags.

Action Type	*TAG*	*Description*	*Action Data*
Move to Folder	MOVE	Move the message to a given E-mail Folder	Folder Path
Copy to Folder	COPY	Insert a copy of the message in a given E-mail Folder	Folder Path
Delete Message	DELETE	Delete the Message	Nothing
Forward Message	FORWARD	Forward the message to a given internet address	Internet Address
Auto Reply	AUTOREPLY	Automatically Reply to the message with a static message	Path to Static Message in RFC822 format
Do not Download	NOTDOWNLOAD	Do not download the message from the server	Nothing
Delete the Message from the Server	DELETESVR	Delete the Message from the Server	Nothing
Replace Message	REPLACE	Replace the Message with a Static Message and existing header	Path to Static Message in RFC822 format
Play Sound	PLAY	Play a Sound	Path to Sound File.
Popup an Alert	POPUP	Popup an Alert	Text to put in alert.
Open E-mail Read Dialog	OPENREAD	Open an E-mail read dialog with the current message loaded	Nothing
Open a chat window to sender	OPENCHAT	Open a Chat window to the Sender of the Message.	Nothing
Report to Abuse	ABUSE	Send the Header to BellSouth E-mail Abuse Center	Nothing
Report as Spam	SPAM	Forward the Message to "thisisspam@bellsouth.net"	Nothing

CHART 2: Rule Actions

[0044] In some embodiments, the rules may be stored on the local system in an XML-based text file. For the embodiments described above, the root node for the XML-based text file is a "RULE" tag (*e.g.*, <RULE . . . >). In those embodiments, the RULE tag has value pairs for rule ID (*e.g.*, ruleID = "001001"), rule type (*e.g.*, ruleType = "System"),

and order (*e.g.*, order = "1"). The order value pair determines the order in which to execute the rule.

[0045] The "CRITERIA" tag (*e.g.*, "<CRITERIA>") and the "ACTION" tag (*e.g.*, "<ACTION>"), which identify the condition and the action, respectively, may be located under the RULE tag. Optionally, an "EXCEPTION" tag may also exist under the RULE tag, thereby providing any exceptions to the rule. Similar to the CRITERIA tag and the ACTION tag, the EXCEPTION tag may be defined by value pairs. The CRITERIA tag describes the condition for which the rule will be executed. The ACTION tag describes the action that will be performed if the CRITERIA is met. The EXCEPTION tag describes the case when the rule will not be executed.

[0046] If multiple CRITERIA tags exist within a rule, then an "operator" value pair may be provided, in order to define whether the conditions should be met in the conjunctive ("and") or in the disjunctive ("or"). In other words, the operator value pair determines how to logically bind the conditions. In some embodiments, if an operator value pair is not supplied, then the default value may be the conjunctive "and" operation. In other embodiments, the default may be set to the "or" operation.

[0047] Thus, for example, a rule may appear as follows:

```
[0048] <RULE ruleID="001001" ruleType="System" order="1">
[0049]     <CRITERIA>
[0050]         <BODYKEY operator="OR" data="XXX"></BODYKEY>
[0051]         <SUBJECTKEY operator="OR" data="XXX"></SUBJECTKEY>
[0052]     </CRITERIA>
[0053]     <EXCEPTION>
[0054]         <FROMADDR data="foo@foo.com"></FROMADDR>
[0055]     </EXCEPTION>
[0056]     <ACTION>
[0057]         <DELETE></DELETE>
[0058]         <SPAM></SPAM>
[0059]     </ACTION>
[0060] </RULE>
```

[0061] In the example rule, the ruleID of 001001 uniquely identifies the rule. The example rule is a system rule, which, for example, is provided by the vendor. Additionally, this rule has an order of "1" (*i.e.*, order="1"), which indicates that this rule should be processed prior to processing other rules.

[0062] In the example rule, the condition (*i.e.*, <CRITERIA>) for performing an action is the text "XXX" (*i.e.*, data="XXX") being found in either the text body (*i.e.*, BODYKEY) of the digital message or (*i.e.*, operator = "OR") the text "XXX" being found in the subject line (*i.e.*, SUBJECTKEY) of the digital message.

[0063] The rule should not be executed if the digital message is received from foo@foo.com (*i.e.*, FROMADDR data="foo@foo.com"). Thus, if either of those conditions are met, and the digital message is not from foo@foo.com, then, for the example rule, the corresponding action results in deletion of the email message (*i.e.*, <DELETE></DELETE>) and reporting of the email as SPAM (*i.e.*, <SPAM></SPAM).

[0064] Having described several embodiments of rule syntax and storage, FIGS. 2 through 6 provide several embodiments of methods for processing rules for digital messages.

[0065] FIG. 2 is a flowchart showing an embodiment of a method for processing rules. As shown in FIG. 1, an embodiment of the process may be seen as comprising the steps of providing (210) a programming interface 185, which permits entry of a condition and a corresponding action by a user. The embodiment of the method further includes the step of storing (220) the condition and action provided by the user at the programming interface 185. In a preferred embodiment, the programming interface 185 may be a text editor at which the user may provide XML-tagged conditions, actions, and exceptions. In this regard, the text editor provides an interface at which the user may input the various

conditions such as those provided in CHART 1 and the various actions such as those provided in CHART 2.

[0066] In another embodiment, the user interface may be one or more graphical user interfaces that query the user for input. In some embodiments, multiple user interfaces are sequentially presented to the user, with each user interface querying the user for a specific piece of information. For example, as shown in FIG. 7, the user interface may be a SPAM filter. The user interface provides user-selectable options to activate or deactivate the function. In the example of FIG. 7, options are provided to either turn "on" or turn "off" the SPAM filtering functions. When the user selects one of the options by, for example, clicking on the selection using a mouse or other pointing device, the underlying software performs the corresponding function by selectively activating or deactivating the filtering function. In some embodiments, in which the rule engine is implemented using XML, the activation or deactivation of the filtering function may be performed by toggling an XML-based value pair (*e.g.*, a tag and its corresponding argument) that corresponds to the filtering function.

[0067] In some embodiments, when the filter is turned "on," additional options for filter settings are provided. For example, options may be provided to create or edit a "block list" or an "allow list." The block list includes email addresses of specific senders from whom the user chooses not to receive any email messages. The allow list includes email addresses of specific senders from whom the user will always receive email messages. Since various example implementations of both the block list and the allow list would be understood by those skilled in the art after reading the present disclosure, including documents incorporated herein by reference, further discussion of the block list and the allow list is omitted here.

[0068] In addition to the block list and the allow list, the sensitivity of the filter may be adjusted. In some embodiments, the filter is implemented as a Bayesian filter, which is known by those having ordinary skill in the art, as evidenced by publications such as, for example, "A Plan for Spam" by Paul Graham, published at http://www.paulgraham.com/spam.html, in August of 2002 (also referred to herein as "the Graham article"), which is incorporated herein by reference in its entirety. As known to those skilled in the art, the sensitivity of the Bayesian filters (or other similar filters) may be varied by assigning various weights to the filtering functions. Since the underlying mechanism for varying of the sensitivity of filters is known in the art, further discussion of the underlying mechanism is omitted here. However, unlike conventional approaches, several embodiments of the present disclosure provide a user-friendly approach to varying the sensitivity of the filter. For example, in conventional approaches, the various weights are directly adjusted by the user, who inputs specific numeric values as weights to the functions.

[0069] In contrast to the conventional approaches, the embodiments of the present disclosure provide a user-friendly interactive interface in which a user is queried in plain English for various settings. For example, rather than providing specific numeric weights, the user is queried for whether the filter should have a "high" sensitivity or a "low" sensitivity. This query may be in the form of a "sliding scale" on a graphical user interface, similar to that shown in FIG. 7. Upon input by the user, the input is converted to a specific numeric value for the user, thereby alleviating the user from performing rigid calculations. In other words, rather than having the user calculate the various weights to the filtering functions, the several embodiments of the disclosure perform the calculation of the weights by correlating the user's input to varying weights. For example, if the

user's input reflects a "high" degree of sensitivity, then the underlying filtering mechanism may, among others, assign a higher numeric value (*e.g.*, 90%) to the weight of the filtering function for undesired words (or vice versa), include additional tokens in the filtering process, assign a lower numeric value (*e.g.*, 10%) to the weight of the filtering function for desired words (or vice versa), *etc.* Conversely, if the user's input reflects a "low" degree of sensitivity, then the underlying filtering mechanism may, among others, assign a more neutral numeric value (*e.g.*, 65%) to the weight of the filtering function for undesired words (or vice versa), include fewer tokens in the filtering process, *etc.* Greater convenience to the user is achieved by providing a user-friendly interface in which the user is alleviated from directly performing complex calculations.

[0070] While a filtering rule has been described in great detail above, it should be appreciated that other rules may be established in a similar manner. For example, user-friendly, plain-English, interactive interfaces may be provided to the user for establishing rules that save messages into various folders. Similarly, for other embodiments, user-friendly interactive interfaces may be provided for establishing rules that launch instant messaging (IM) chat sessions with email senders. These, and various other functions, are shown with reference to FIGS. 3 through 6.

[0071] For rules that are written in XML and stored in an XML database, it should be appreciated that the rules, once established and stored, may be accessed by a user through, for example, a text editor. Alternatively, the rules may, in other embodiments, be accessed by a user through a menu-driven mechanism. Since text editors and menu-driven mechanisms are known in the art, further discussion of such mechanisms and editors is omitted here. Once accessed, the user may selectively edit, delete, rename, *etc.* the rules as desired.

[0072] FIG. 3 is a flowchart showing another embodiment of a method for processing rules. The embodiment of FIG. 3 shows a process that begins after one or more rules have been created and stored. In this regard, the process of FIG. 3 presumes that predefined rules already exist in the system. These predefined rules may be various permutations of conditions and actions, as shown in CHART 1 and CHART 2. Thus, the embodiment of FIG. 3 begins when a digital message, such as an email message, is received (310). Upon receiving (310) the digital message, the system determines (320) whether or not the digital message meets the predefined condition. If the digital message meets the predefined condition, then the process ends. If, on the other hand, the digital message does not meet the predefined condition, then a filtering algorithm is executed (330) on the digital message. Thus, FIG. 3 provides an example in which a filtering algorithm is executed (330) unless there is some indication to prevent execution of the filtering algorithm. For example, if an email application receives an email message from foo@foo.com, and email from that sender is always welcome, then that email message will be received without further filtering.

[0073] FIG. 4 is a flowchart showing, in greater detail, the step of determining (320) whether or not a digital message meets a predefined condition, as shown in FIG. 3. Specifically, the embodiment of FIG. 4 provides an example of determining (320) whether or not a received email message should bypass a filter, such as, for example, a Bayesian filter.

[0074] As shown in the embodiment of FIG. 4, the determining (320) step may begin by first determining (405) whether or not an email message has an empty subject line. If the email message has an empty subject line, then the process exits to FIG. 3, and a filtering algorithm is executed (330) on the email message. If, on the other hand, the subject line

is not empty, then the process continues by next determining (410) whether or not the message body is empty. If the message body is empty, then the process exits to FIG. 3, and the filtering algorithm is executed (330) on the email message. Conversely, if the message body is not empty, then the process next determines (415) whether or not the size of the message is greater than a predefined size. In some embodiments, the threshold for email size may be two or three megabytes. It should, however, be appreciated that this threshold may be varied according to the various needs of the user. If the message size exceeds the predefined threshold, then the process exits to FIG. 3, and the filtering algorithm is executed (330) on the email message. If, however, the threshold message size is not exceeded, then the process continues by extracting (420) various features from the email message. The various features may include the Internet address of the sender, the Internet address of the recipient, Internet domain names, words in the subject line of the message, words in the body of the message, HTML or XML tags in the email message, IP addresses of intermediate Internet hops, or a variety of other features. Since these features are discussed in greater detail in 10/610,736, filed on June 30, 2003, [attorney docket no. 190250-1300], titled "Identifying Undesired Email Messages Having Attachments," filed on October 14, 2003, and [attorney docket no. 190250-1610], titled "Phonetic Filtering of Undesired Email Messages," filed on October 14, 2003, further discussion of these features is omitted here. Upon extracting (420) the various features, the features are compared (425) with a predefined list of features, and the system determines (430) whether or not the extracted feature exists in the predefined list. If the feature does not exist in the predefined list, then the process exits to FIG. 3, and the filtering algorithm is executed (330) on the email message. Alternatively, if the extracted

feature exists in the predefined list, then the process ends without additionally filtering the email message.

[0075] For example, if the user does not wish to additionally filter an email message from foo@foo.com, then foo@foo.com will be an entry in the predefined list. Thus, if the extracted Internet address of the sender is foo@foo.com, then the additional filtering algorithm is not executed on that email message.

[0076] FIG. 5 is a flowchart showing yet another embodiment of a method for processing rules. The embodiment of FIG. 5 shows a process that begins after one or more rules have been created and stored. In this regard, the process of FIG. 5 presumes that predefined rules already exist in the system. These predefined rules may be various permutations of conditions and actions, as shown in CHART 1 and CHART 2. Thus, the embodiment of FIG. 5 begins when a digital message, such as an email message, is received (510) from a sender. Upon receiving the email message, contact information of the sender is extracted (520) from the email message. The contact information may be the email address of the sender, the name of the sender, or other information indicative of the sender. The extracted (520) contact information is compared (530) with a previously-stored list of contacts, and the system determines (540) whether or not the contact information is stored in that list. If the contact information is not stored in that list, then the process ends. If, however, the contact information exists in the list, then the system further determines (550) whether or not the sender is present online (*e.g.*, present and available). Since the determination of the sender's presence from the sender's email contact information is described in greater detail in U.S. patent application serial number 10/325,268 and U.S. patent application serial number 10/274,408, further discussion of determining (550) the sender's presence is omitted here. If the system determines (550)

that the sender is not present online, then the process ends. Conversely, if the system determines that the sender is present online, then an IM chat session is initiated (560) between the recipient and the sender.

[0077] FIG. 6 is a flowchart showing a specific embodiment of another rules-processing method. The embodiment of FIG. 6 shows a process that begins after one or more rules have been created and stored. In this regard, the process of FIG. 6 presumes that predefined rules already exist in the system. These predefined rules may be various permutations of conditions and actions, as shown in CHART 1 and CHART 2. Thus, the embodiment of FIG. 6 begins when a digital message, such as an email message, is received (605) from a sender. Upon receiving (605) the digital message, the system determines (610) whether or not the digital message contains a command. The command may be a text string in the message, such as, for example, "get file." If the digital message does not contain a command, then the process ends. If, however, the digital message does contain a command, then the system further determines (615) whether or not the command is associated with an argument (*e.g.*, file name). For example, the message may contain a text string such as "get file = foo.doc." If the command (*e.g.*, "get file") is not associated with an argument (*e.g.*, file name, "foo.doc"), then an error message is generated (620), which indicates that there is no argument for the command. The generated (620) error message is transmitted (625) to the sender of the digital message, after which the process is terminated. If, in this example, the command is associated with a file name, then the file name is extracted (630). Using the extracted (630) file name, the local data storage devices (*e.g.*, hard drives) are searched (635), and the system determines (640) whether or not such a file exists on the local hard drives. If the file does not exist locally, then an error message is generated (645), which indicates

that the file could not be found. The error message is transmitted (625) to the sender of the digital message, and the process is terminated. If the requested file is found locally, then the system further determines (650) whether or not access to the file has been restricted. If access to the file has been restricted by, for example, defining the file property as "hidden" or "private," then an error message is generated (655), which indicates that the file is not accessible. That error message is transmitted (625) to the sender of the digital message, and the process is thereafter terminated. If the requested file is accessible, then the file is retrieved (660) and transmitted (665) to the sender of the digital message. In this regard, as shown in the embodiment of FIG. 6, the rule processing method may be customized to carry out a variety of functions previously unavailable in conventional email and IM applications.

[0078] As shown in the various embodiments above, by providing a versatile rule engine, the functionality for both email and IM applications is increased.

[0079] The email application 181, the IM application 183, the programming interface 185, and the command execution logic 187 may be implemented in hardware, software, firmware, or a combination thereof. In the preferred embodiment(s), the email application 181, the IM application 183, the programming interface 185, and the command execution logic 187 are each implemented in software or firmware that is stored in a memory and that is executed by a suitable instruction execution system. If implemented in hardware, as in an alternative embodiment, the email application 181, the IM application 183, the programming interface 185, and the command execution logic 187 can be implemented with any or a combination of the following technologies, which are all well known in the art: a discrete logic circuit(s) having logic gates for implementing logic functions upon data signals, an application specific integrated circuit

(ASIC) having appropriate combinational logic gates, a programmable gate array(s) (PGA), a field programmable gate array (FPGA), etc. In this regard, it should be appreciated that the IM application 183 may include presence-determination logic, IM-chat-initiation logic, and other structures that are specifically configured to carry out relevant IM functions. Similarly, it should be appreciated that the email application 181 may include condition-determination logic, information-extraction logic, and other structures that are specifically configured to carry out relevant email functions. Likewise, it should be appreciated that the programming interface 185 may include program-interface logic, which provides the structural components that are configured to render a user interface to receive user input, and other relevant structures that are specifically configured to carry out the various functions of the programming interface 185.

[0080] Any process descriptions or blocks in flow charts should be understood as representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process, and alternate implementations are included within the scope of the preferred embodiment of the present disclosure in which functions may be executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those reasonably skilled in the art of the present disclosure.

[0081] The email application 181, the IM application 183, the programming interface 185, and the command execution logic 187 may be computer programs, which comprise ordered listings of executable instructions for implementing logical functions. As such, these programs may be embodied in any computer-readable medium for use by or in connection with an instruction execution system, apparatus, or device, such as a

computer-based system, processor-containing system, or other system that can fetch the instructions from the instruction execution system, apparatus, or device and execute the instructions. In the context of this document, a "computer-readable medium" can be any means that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device. The computer-readable medium can be, for example but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, device, or propagation medium. More specific examples (a nonexhaustive list) of the computer-readable medium would include the following: an electrical connection (electronic) having one or more wires, a portable computer diskette (magnetic), a random access memory (RAM) (electronic), a read-only memory (ROM) (electronic), an erasable programmable read-only memory (EPROM or Flash memory) (electronic), an optical fiber (optical), and a portable compact disc read-only memory (CDROM) (optical). Note that the computer-readable medium could even be paper or another suitable medium upon which the program is printed, as the program can be electronically captured via, for instance, optical scanning of the paper or other medium, then compiled, interpreted or otherwise processed in a suitable manner if necessary, and then stored in a computer memory.

[0082] Although exemplary embodiments have been shown and described, it will be clear to those of ordinary skill in the art that a number of changes, modifications, or alterations to the disclosure as described may be made. All such changes, modifications, and alterations should therefore be seen as within the scope of the disclosure.